

SCHOOL OF EECE

ECE107L/A13

SIGNALS SPECTRA, AND SIGNAL PROCESSING

PROJECT

**Submitted by:**

**GROUP 3**

**DOCDOCIL, CHRISTIAN G.**

**VILLENA, EDUARDO B.**

Engr. Leonardo D. Valiente

Instructor

**INTRODUCTION OF THE PROJECT**

Modern technology now enable us to do a lot of different things. While many other programs offer features that lets us do these things easily, it can sometimes be beneficial to us that we learn how do these programs work. One of these programs is Photoshop which is developed by Adobe. It is a powerful software that allows us to edit images and make modifications that is before is complicated and difficult to do. With this application, innovation simply took place and lots of other programs were also created similar to that of Photoshop.

# **ACKNOWLEDGMENT**

Our deepest gratitude to our Engr. Leonardo D. Valiente Jr. for whole-heartedly teaching and guiding us about digital signal processing. His teachings made this project a possibility as the professor of our digital signal processing laboratory class.

**DOCDOCIL, Christian G.**

**VILLENA, Eduardo B.**

# **PROGRAM DESCRIPTION**

The program is developed under MATLAB R2016b with the Image Processing Toolbox. It retrieves an image file with a .jpg, .tif, .png, and .gif. It has the capability to do FIR filter image processing, such as motion, gaussian, average, unsharp, laplacian, and sobel. The input image can also be converted to binary or grayscale. Morphological conversions such as invert, erosion, dilate, estimate, and subtract are also in the program. A contrast slider is also present in the program, which has an option of low in, low out, high in, or high out. Upon opening an input image, a histogram is automatically generated. This includes the red, green, and blue spectrum. It is also automatically generated after using an option and then clicking the apply button and after configuring the generated image it can be save using file menu and ‘Save Generated Image’.

# **SCOPE AND LIMITATIONS**

The program is only limited to image files that are locally stored in the machine. It cannot retrieve an image located on the internet and download it on the host machine. Apart from the provided options in the program itself, it cannot do anything more.

# **SCREEN CAPTURE OF IMAGE FILE**

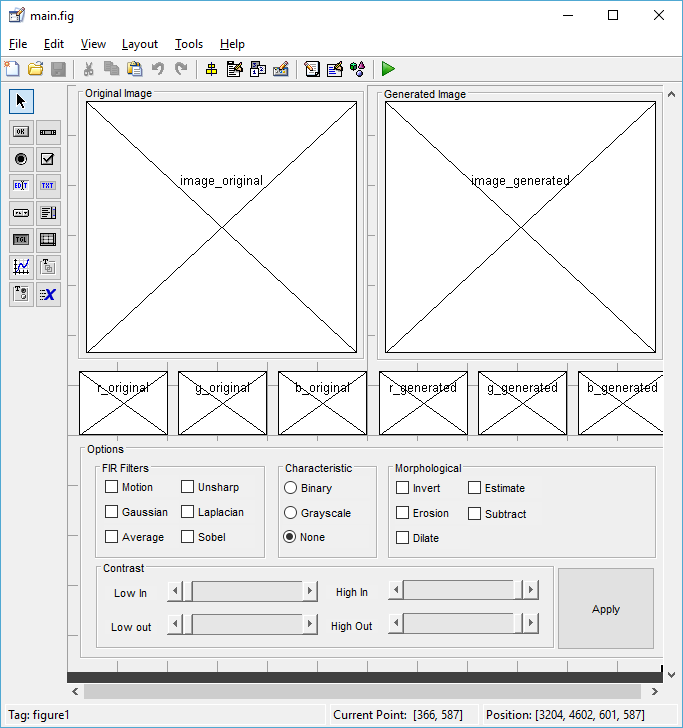


Fig. 1) Image file screen capture

# **ALL M-FILES USED IN THE PROJECT**

function varargout = main(varargin)

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @main\_OpeningFcn, ...

'gui\_OutputFcn', @main\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

function main\_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);

movegui('center');

function varargout = main\_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

% --------------------------------------------------------------------

function menu\_file\_Callback(hObject, eventdata, handles)

% --------------------------------------------------------------------

function menu\_other\_Callback(hObject, eventdata, handles)

% --------------------------------------------------------------------

function other\_help\_Callback(hObject, eventdata, handles)

% --------------------------------------------------------------------

function file\_open\_Callback(hObject, eventdata, handles)

[filename,pathname] = uigetfile({'\*.jpg;\*.tif;\*.png;\*.gif','All Image Files';'\*.\*','All Files' },'Open an image...');

global full\_path

full\_path = strcat(pathname,filename);

disp(full\_path);

axes(handles.image\_original);

imshow(full\_path);

image\_original = imread(full\_path);

Red = image\_original(:,:,1);

Green = image\_original(:,:,2);

Blue = image\_original(:,:,3);

axes(handles.r\_original);

histogram(Red,'FaceColor','r','EdgeColor','r');

axis off

axes(handles.g\_original);

histogram(Green,'FaceColor','g','EdgeColor','g');

axis off

axes(handles.b\_original);

histogram(Blue,'FaceColor','b','EdgeColor','b');

axis off

apply\_button\_Callback(handles.apply\_button,eventdata,handles);

% --------------------------------------------------------------------

function file\_save\_Callback(hObject, eventdata, handles)

global image\_filtered

imwrite(image\_filtered,'result.jpg');

% --- Executes on button press in checkbox1.

function checkbox1\_Callback(hObject, eventdata, handles)

% --- Executes on button press in apply\_button.

function apply\_button\_Callback(hObject, eventdata, handles)

global full\_path;

global image\_filtered;

global image\_original;

image\_original = imread(full\_path);

image\_filtered = imread(full\_path);

low\_in = get(handles.c\_lowin,'value');

low\_out = get(handles.c\_lowout,'value');

high\_in = get(handles.c\_highin,'value');

high\_out = get(handles.c\_highout,'value');

disp(low\_in);

disp(low\_out);

disp(high\_in);

disp(high\_out);

image\_filtered = imadjust(image\_filtered, [low\_in;high\_in],[low\_out;high\_out]);

if(get(handles.c\_binary, 'value') == 1)

image\_filtered = rgb2gray(image\_filtered);

image\_filtered = imbinarize(image\_filtered);

elseif(get(handles.c\_grayscale, 'value') == 1)

image\_filtered = rgb2gray(image\_filtered);

else

%do nothing

end

if(get(handles.c\_motion, 'value') == 1)

h = fspecial('motion');

image\_filtered = imfilter(image\_filtered,h);

end

if(get(handles.c\_gaussian, 'value') == 1)

h = fspecial('gaussian',[1000,1],1);

image\_filtered = imfilter(image\_filtered,h);

end

if(get(handles.c\_laplacian, 'value') == 1)

h = fspecial('average');

image\_filtered = imfilter(image\_filtered,h);

end

if(get(handles.c\_disk, 'value') == 1)

h = fspecial('unsharp');

image\_filtered = imfilter(image\_filtered,h);

end

if(get(handles.c\_prewitt, 'value') == 1)

h = fspecial('laplacian');

image\_filtered = imfilter(image\_filtered,h);

end

if(get(handles.c\_sobel, 'value') == 1)

h = fspecial('sobel');

image\_filtered = imfilter(image\_filtered,h);

end

if(get(handles.c\_invert, 'value') == 1)

image\_filtered = imcomplement(image\_filtered);

end

if(get(handles.c\_erosion, 'value') == 1)

image\_filtered = imerode(image\_filtered,ones(3));

end

if(get(handles.c\_erosion, 'value') == 1)

image\_filtered = imerode(image\_filtered,ones(3));

end

if(get(handles.c\_estimate, 'value') == 1)

image\_filtered = imopen(image\_filtered,strel('disk',15));

end

if(get(handles.c\_subtract, 'value') == 1)

image\_filtered = imsubtract(image\_original,image\_filtered);

end

axes(handles.image\_generated);

imshow(image\_filtered);

if(get(handles.c\_grayscale, 'value') == 1 || get(handles.c\_binary, 'value') == 1 )

cla(handles.r\_generated);

cla(handles.b\_generated);

axes(handles.g\_generated);

imhist(image\_filtered);

else

Red = image\_filtered(:,:,1);

Green = image\_filtered(:,:,2);

Blue = image\_filtered(:,:,3);

axes(handles.r\_generated);

histogram(Red,'FaceColor','r','EdgeColor','r');

axis off

axes(handles.g\_generated);

histogram(Green,'FaceColor','g','EdgeColor','g');

axis off

axes(handles.b\_generated);

histogram(Blue,'FaceColor','b','EdgeColor','b');

axis off

end

% --- Executes on selection change in popupmenu1.

function popupmenu1\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array

% contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.

function popupmenu1\_CreateFcn(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in reset\_button.

function reset\_button\_Callback(hObject, eventdata, handles)

% hObject handle to reset\_button (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global image\_original

global image\_filtered

figure(2),imhist(rgb2gray(image\_original));

% --- Executes on button press in c\_laplacian.

function c\_laplacian\_Callback(hObject, eventdata, handles)

% hObject handle to c\_laplacian (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_laplacian

% --- Executes on button press in c\_gaussian.

function c\_gaussian\_Callback(hObject, eventdata, handles)

% hObject handle to c\_gaussian (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_gaussian

% --- Executes on button press in c\_motion.

function c\_motion\_Callback(hObject, eventdata, handles)

% hObject handle to c\_motion (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_motion

% --- Executes on button press in c\_sobel.

function c\_sobel\_Callback(hObject, eventdata, handles)

% hObject handle to c\_sobel (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_sobel

% --- Executes on button press in c\_prewitt.

function c\_prewitt\_Callback(hObject, eventdata, handles)

% hObject handle to c\_prewitt (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_prewitt

% --- Executes on button press in c\_disk.

function c\_disk\_Callback(hObject, eventdata, handles)

% hObject handle to c\_disk (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_disk

function c\_contrast\_Callback(hObject, eventdata, handles)

% hObject handle to c\_contrast (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c\_contrast as text

% str2double(get(hObject,'String')) returns contents of c\_contrast as a double

% --- Executes during object creation, after setting all properties.

function c\_contrast\_CreateFcn(hObject, eventdata, handles)

% hObject handle to c\_contrast (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

function c\_threshold\_Callback(hObject, eventdata, handles)

% hObject handle to c\_threshold (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c\_threshold as text

% str2double(get(hObject,'String')) returns contents of c\_threshold as a double

% --- Executes during object creation, after setting all properties.

function c\_threshold\_CreateFcn(hObject, eventdata, handles)

% hObject handle to c\_threshold (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

% See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor','white');

end

% --- Executes on button press in c\_invert.

function c\_invert\_Callback(hObject, eventdata, handles)

% hObject handle to c\_invert (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_invert

% --- Executes on button press in c\_erosion.

function c\_erosion\_Callback(hObject, eventdata, handles)

% hObject handle to c\_erosion (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_erosion

% --- Executes on button press in c\_dilate.

function c\_dilate\_Callback(hObject, eventdata, handles)

% hObject handle to c\_dilate (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_dilate

% --- Executes on button press in c\_estimate.

function c\_estimate\_Callback(hObject, eventdata, handles)

% hObject handle to c\_estimate (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_estimate

% --- Executes on button press in c\_subtract.

function c\_subtract\_Callback(hObject, eventdata, handles)

% hObject handle to c\_subtract (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of c\_subtract

% --- Executes on slider movement.

function c\_lowin\_Callback(hObject, eventdata, handles)

% hObject handle to c\_lowin (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.

function c\_lowin\_CreateFcn(hObject, eventdata, handles)

% hObject handle to c\_lowin (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor',[.9 .9 .9]);

end

% --- Executes on slider movement.

function slider3\_Callback(hObject, eventdata, handles)

% hObject handle to slider3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.

function slider3\_CreateFcn(hObject, eventdata, handles)

% hObject handle to slider3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor',[.9 .9 .9]);

end

% --- Executes on slider movement.

function c\_lowout\_Callback(hObject, eventdata, handles)

% hObject handle to c\_lowout (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.

function c\_lowout\_CreateFcn(hObject, eventdata, handles)

% hObject handle to c\_lowout (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor',[.9 .9 .9]);

end

% --- Executes on slider movement.

function c\_highin\_Callback(hObject, eventdata, handles)

% hObject handle to c\_highin (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.

function c\_highin\_CreateFcn(hObject, eventdata, handles)

% hObject handle to c\_highin (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor',[.9 .9 .9]);

end

% --- Executes on slider movement.

function c\_highout\_Callback(hObject, eventdata, handles)

% hObject handle to c\_highout (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider

% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.

function c\_highout\_CreateFcn(hObject, eventdata, handles)

% hObject handle to c\_highout (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))

set(hObject,'BackgroundColor',[.9 .9 .9]);

end

% --- Executes on button press in c\_edited.

function c\_edited\_Callback(hObject, eventdata, handles)

% hObject handle to c\_edited (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

# **THREE (3) SAMPLE OUTPUT**

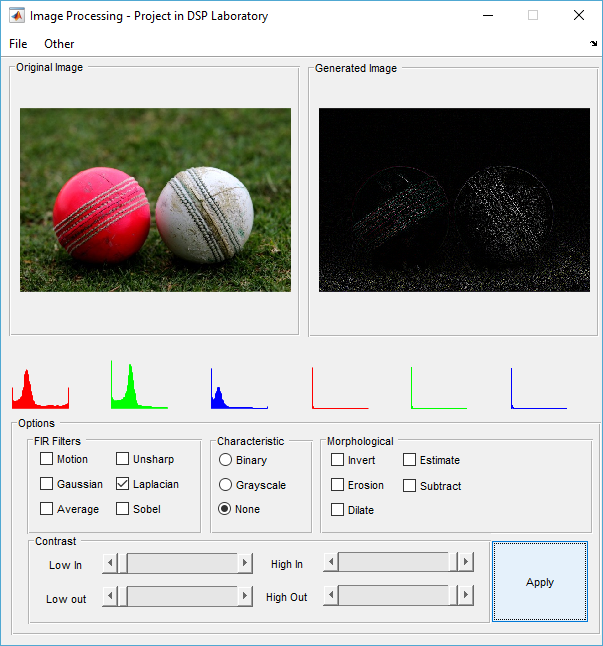


Fig. 2) First Output

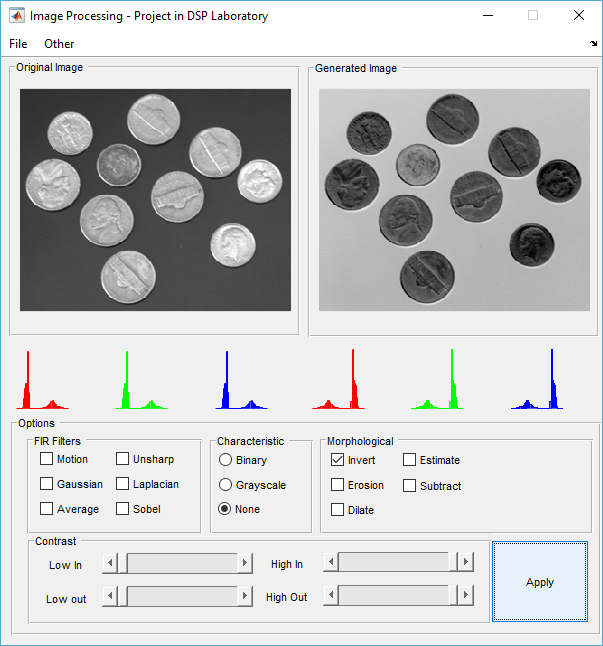


Fig. 3) Second Output

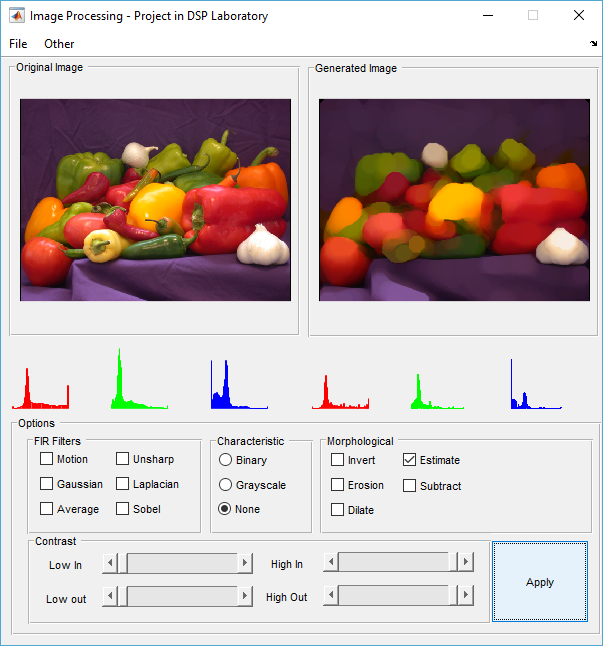


Fig. 4) Third Output

# **REFERENCES**

[1] Image Processing Toolbox. (n.d.). Retrieved March 05, 2017, from https://www.mathworks.com/products/image.html

[2] fspecial - MATLAB. (n.d.). Retrieved March 05, 2017, from https://www.mathworks.com/help/images/ref/fspecial.html

[3] Morphologically open image - MATLAB imopen. (n.d.). Retrieved March 05, 2017, from https://www.mathworks.com/help/images/ref/imopen.html

[4] Perform morphological opening on an image - MATLAB. (n.d.). Retrieved March 05, 2017, from https://www.mathworks.com/help/vision/ref/vision.morphologicalopen-class.html

[5] How to calculate a RGB histogram? (n.d.). Retrieved March 05, 2017, from https://www.mathworks.com/matlabcentral/newsreader/view\_thread/68712